

DIPLOMAMUNKA

Szilágyi Attila

Debrecen

2010

Debreceni Egyetem
Informatika Kar

INTERAKTÍV INFORMÁCIÓS PANEL KÉPFELDOLGOZÓ ALGORITMUSAI

Témavezető:
Dr. Fazekas Attila
egyetemi docens

Készítette:
Szilágyi Attila
programtervező matematikus

Debrecen
2010

Tartalomjegyzék

1. Bevezetés	5
1.1. A fejlesztés során használt eszközök	6
2. Irodalmi áttekintés	7
2.1. Objektumdetektálás	7
2.2. Az arc vizsgálata	8
3. Arcdetektálás	9
3.1. Az integrált kép	10
3.2. A betanítás	10
3.3. Kaszkád struktúra	11
4. Az előfeldolgozás	13
4.1. Az arc kivágása	13
4.2. Local Binary Patterns	13
4.3. Személyfüggetlen érzelemfelismerés Local Binary Patterns segítségével . . .	16
4.4. Érzelemfelismerés boosting Local Binary Patterns-el	17
4.5. A biológiai nem felsimerése boosting Local Binary Patterns-el	18
5. Számítógépes tanulás	20
5.1. AdaBoost	20
5.2. Support Vector Machine	20
5.3. Incremental Support Vector Machine	21
6. A rendszer részletes bemutatása	23
6.1. A rendszer felépítése	23
6.2. A FaceClassifier dinamikus könyvtár	23
6.3. A konfigurációs állományok	24
6.4. A vibrálás kiküszöbölése	25
6.5. A tanuláshoz és a tesztekhez kialakított adatbázisok	27

6.6. Az egyes adatbázisok	27
6.7. Eredmények	29
6.8. Konzekvencia	32
7. A rendszer működés közben	33
8. Összefoglalás	35
9. Irodalomjegyzék	36
10.Köszönetnyilvánítás	39

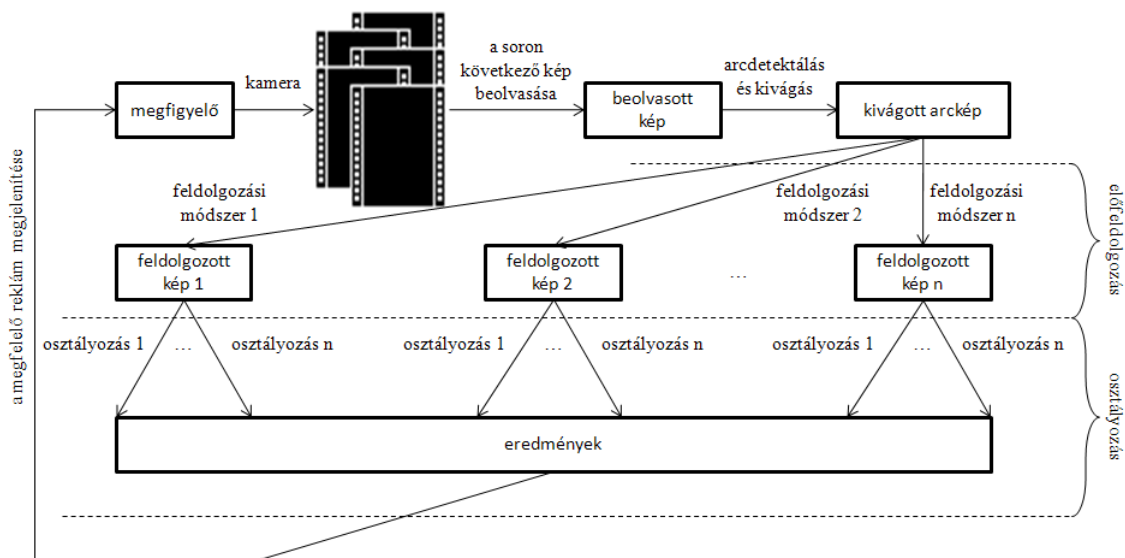
1. Bevezetés

A számítógéppel történő képfeldolgozás egy igen szerteágazó területe az informatikának. Ezen belül az emberi arc vizsgálatához kötődő automatikus érzelem felismerés, életkor és biológiai nem meghatározása vagy a különböző kulturális csoportokba történő besorolás érdekes és komoly kihívásokat rejtő részterület. Ezen felül az itt szerzett tudás hasznosíthatónak ígérkezik például a biztonsági iparban, a pszichológiában vagy akár a szórakoztatásban is.

Jelen dolgozat tárgya egy olyan interaktív reklámfelületnek a kialakítása, amely személyre szabott termékkínálattal próbál bennünket vásárlásra ösztönözni. Ennek során egy, a reklámfelülethez tartozó kamera látja el alkalmazásunkat képi adattal, amiből az esetleges vásárlóra vonatkozó, lehető legtöbb és legpontosabb információt szeretnénk megállapítani. A rendszer a kamera és egyúttal a hirdetőtábla előtt álló személy külső tulajdonságai alapján ajánl olyan reklámot, mely leginkább felkeltheti az érdeklődését.

Nyilvánvaló, hogy a különböző demográfiai csoportok eltérő vásárlói preferenciákkal rendelkeznek, így a megjelenítésre kerülő hirdetés kiválasztásához a hirdetőfelületet megfigyelő személy bizonyos tulajdonságainak minél pontosabb meghatározására van szükség. Ennek értelmében ki kell választanunk melyek azok a lényeges szempontok, melyek alapján a vásárlókat kategorizálni szeretnénk (pl.: nem, életkor, ...), majd az adott kategóriában praktikusán alkalmazandó osztályokat kell meghatároznunk (az életkor esetében pl.: fiatal / középkorú / idős). Látható, hogy minden kategória esetén osztályozásról beszélhetünk.

A személy egy adott szempont szerinti osztályába sorolásának folyamata több, az 1. ábrán látható részfeladatra tagolódik: a személyt tartalmazó kép beolvasása, a személy megtalálása a képen, az arc kivágása, majd a kivágott kép megfelelő előfeldolgozását követő osztályozása történik. Fő célom ezen utolsó résztvevékenység (előfeldolgozás + osztályozás) minél hatékonyabb és megbízhatóbb megvalósítása volt. Ugyanakkor hangsúlyozni kell, hogy az arc megtalálása feltétele bármilyen további vizsgálatnak, ezért ez a részfeladat kiemelkedő fontosságú: csak az arc sikeres detektálása után nyílik lehetőségünk annak feldolgozására.



1. ábra. A feldolgozás folyamatának lépései.

1.1. A fejlesztés során használt eszközök

A fejlesztést a Microsoft Visual C++ 2008 Express Edition fejlesztő környezet alatt végeztük, melynek során a képfeldolgozó rutinokhoz az OpenCV, az XML műveletekhez a Libxml2 és a szálkezeléshez a Pthread könyvtárcsomagot alkalmaztuk. A fejlesztés alapvetően Windows XP operációs rendszeren zajlott, de a kódolás során platformfüggetlen megoldásokat alkalmaztunk.

2. Irodalmi áttekintés

2.1. Objektumdetektálás

Az objektumdetektálás feladata, hogy egy képről megállapítsa, hogy tartalmaz-e egy adott objektumot vagy sem, és pozitív válasz esetén megadja az objektum helyét és kiterjedését. Az objektumdetektálásra számos, jellegében eltérő módszer használatos, mint például a megjelenés-alapú (appearance based) [20, 8], tudás-alapú (knowledge based) vagy bizonyos mintaillesztő eljárások (template matching). Ezek közül a leghatékonyabbnak a megjelenés-alapú bizonyult, melynek során egy úgynevezett pozitív és egy negatív képhalmazt használunk fel a tanítás során. Több különböző osztályozási technika hozott kiváló eredményeket (neurális hálózatok, SVM).

Egy objektumdetektorral támasztott legfontosabb kritériumok a következők: jó detektálási ráta (kevés tévesen elvetett vagy tévesen felismert arc), pontosság, megfelelő gyorsaság, robosztusság (azaz meg kell birkóznia az elforgatással és a különböző fényviszonyokkal).

A mi esetünkben a detektálandó objektum egy arc lesz, így a tanítás során a pozitív halmaz arcképeket, a negatív halmaz pedig olyan képeket tartalmaz, amelyeken az arcon kívül bármi szerepelhet.

A kaszkádolt arc detektor a különböző felbontásokban próbálja megtalálni az arcot a bemeneti kép bal felső sarkától kezdve a jobb alsóig. Minden lépésben a képnek csak egy kisebb részét vizsgáljuk, így az arcdetektálási problémát egy arcosztályozás problémára vezetjük vissza. A kaszkádolt osztályozó gyenge osztályozók (tulajdonképpen Haar-szerű sajátosságok) egy halmazából áll, amelyeket az AdaBoost segítségével választunk ki. A hatékonyságot az úgynevezett integrált képen alapuló belső reprezentáció lényegesen növeli.

A textúra leírás egy fontos eszköze az Ojala és mások [10] által bemutatott LBP operátor, amely az arcok reprezentációja során is hatékony eszköznek bizonyult. Arról van szó, hogy egy kép összes képpontját felcímkézzük az adott képpontot körülvevő pixelek intenzitásainak alapján. A címkéket aszerint határozzuk meg, hogy a környező pixelek közül melyek voltak legalább akkora intenzitásúak, mint az éppen címkézendő. Az így adódó összes lehetséges kombinációt egy bináris számmal jelölhetünk. Elkészítjük az iménti címkék hisztogramját (vagy hisztogramjait), majd ezt használjuk az osztályozás során. [20] kaszkádolt arc detektorához hasonló detektort használtak demográfiai vizsgálatokhoz, melyek során a rasszokat és a nemeket vizsgálták [15]. [1]-ben az LBP reprezentációt alkalmazták arc felismerésre.

Megjegyzendő, hogy egy általános adatbázis hiánya sok esetben sajnos meglehetősen nehezé teszi az eltérő körülmények között született eredmények összevetését. Elég mindössze az eltérő fényviszonyokra gondolnunk.

A különböző megközelítésű arcdetektorokról [7, 25] alapján tájékozódhatunk.

2.2. Az arc vizsgálata

Az embereknél a biológiai nem felismerése egy nagyon fontos, hatásos, gyorsan és már gyermekkorban megtanult képesség. A kutatások eredményeit összegezve levonhatjuk azt a lényeges következtetést, hogy a nem felismerése leginkább a szem régióin és az arc körvonalán alapszik, továbbá lényeges az arc textúrája is [3, 23]. Az életkor meghatározása is hasonlóan történik: ugyancsak meghatározó a koponya formája, illetve az arc textúrája [19].

Ugyan sok különbséget tudnánk felhozni, viszont ezek túl általánosak és nehezen mérhetőek, így nehéz lenne olyan explicit szabályokat találnunk, amely lehetővé tenné egy személy nemének a megállapítását. Ennek következményeként a mintaillesztő vagy a tudás alapú technikákkal szemben, itt is a megjelenés-alapú módszereket részesítjük előnyben.

Az arcok osztályozásának megközelítései leginkább az előfeldolgozásban és a sajátság vektor kialakításában térnek el. Az előfeldolgozás rendszerint a kivágásból, skálázásból és hisztogram kiegyenlítésből áll, a sajátság vektor számolásánál pedig Haar transzformációt, Gabor transzformációt, vagy az LBP-k módszerét alkalmazzuk.

A számítógéppel történő biológiai nem felismerésére szánt első kísérletek a neurális hálózatokra épültek. Gollomb és mások [6] egy kétrétegű teljesen előre-kapcsolt neurális hálózatot alkalmaztak, amit SEXNET-nek neveztek el, és amellyel 30x30-as képeket vizsgáltak. Brunelli és mások [4] HyperBF hálózatokat használtak a férfiak és a nők megkülönböztetésére, melynek során két egymással versengő RBF hálózatot tanítottak be számos geometrikus sajátság segítségével. Moghaddam és mások [24] SVM-mel vizsgáltak alacsony felbontású arcképeket. Wu és mások [22] bevezettek egy valósidejű, AdaBoost-on alapuló, a biológiai nem felismerésére alkalmas rendszert, ahol LUT típusú gyenge osztályozókat tanítottak a Simple Direct Appearance Model módszer segítségével.

3. Arcdetektálás

Láttuk, hogy a kitűzött cél megvalósításához, sok egymásra épülő részfeladat egymás utáni elvégzésére van szükség. Ezek mindegyikének sikeres végrehajtásától nagyban függ a végeredmény, de hangsúlyozandó, hogy az arc sikeres detektálása a képen kulcsfontosságú, ugyanis meghiúsulása, vagy téves detektálás esetén a további lépések értelmetlenek. Ezért szükséges, hogy jól működő arcdetektort használjunk. Az általunk választott módszert az alábbiakban részletesen tárgyaljuk.

Az arcnak fontos szerepe lehet a két ember közötti vagy a számítógép és az ember közötti interakció során. Több dologra következtethetünk egy arcképről: életkorra, nemre vagy például érzelmi állapotra, esetleg beazonosíthatjuk a képen tartózkodó személyt és így tovább. A rendszerünk az eredetileg Viola, P. és Jones, M. [20] által kifejlesztett, majd Lienhart, R. és Maydt, J. által továbbfejlesztett és az OpenCV által implementált arcdetektálási technikát alkalmazza [8]. Ez egy kaszkádolt boosting technika, amely a korábbi, közvetlenül a pixelértékeket felhasználó objektumdetektáló algoritmusoknál lényegesen gyorsabb, és szinte ugyanolyan megbízható. Egy rendkívül gyors (15 frame / sec) és jó - gyakorlatilag az eddig publikált legjobb eredményekkel ekvivalens - arcdetektálási rátával rendelkező módszerről beszélünk. Ez három új, alapvetően fontos ötletnek köszönhető:

- a technika által alkalmazott, a sajátságok nagyon gyors kiszámolhatóságát biztosító belső reprezentációnak (az ún. integrált képnek),
- az AdaBoost-on alapuló tanulási algoritmusnak, mely segítségével csak a legjobb szeparálást lehetővé tevő sajátságokat használjuk fel, így jelentősen csökkentjük az osztályozónk méretét,
- továbbá a módszer kaszkád-struktúrájú felépítésének, amely lehetővé teszi a számunkra közömbös háttér régiók rövid időn belül történő kiszűrését, melynek következtében csak az objektum detektálhatósága szempontjából ígéretesebb régióknál növekszik meg a számításigény.

Több dolog is indokolja, hogy a közvetlen pixelhasználat helyett az egyszerű, képi struktúrákat jól leíró Haar-szerű sajátságokat használjuk: egyrészt a sajátság alapú rendszerek gyorsabbak a pixel alapúaknál, másrészt ezen sajátságok olyan "alaki" információt hordoznak, melynek megtanulása pixelenként, véges tanítóadat esetén meglehetősen nehéz feladat lenne. A Haar-szerű sajátságok példányai úgy számíthatók ki, hogy az adott sajátsághoz tartozó fekete területen található pixelértékek összegéből kivonjuk a fehér területen található pixelértékek összegét. Ez a módszer szűrkeskálás képekre lett kife-

jlesztve, így sok korábban publikált módszertől eltérően nem használ olyan kiegészítő információkat, mint például a bórszín.

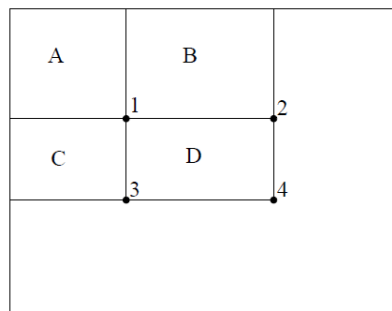
3.1. Az integrált kép

Az eredeti bejövő képeket integrált képpé kell transzformálnunk. Ez lényegében egy olyan közbülső ábrázolás, melynek előnye, hogy néhány pixelenkénti művelettel kiszámolható és a további feldolgozást jelentősen gyorsítja. Az integrált kép bármely (x, y) pozíciójában, az eredeti, transzformálandó kép adott koordinátájától felfele, és balra vett pixelértékeinek összege található:

$$s(x, y) = s(x, y - 1) + i(x, y),$$

$$ii(x, y) = ii(x - 1, y) + s(x, y),$$

ahol ii a transzformált kép, i az eredeti kép, s a növekvő sorösszeg, továbbá $s(x, -1) = 0$ és $ii(-1, y) = 0$. Ezt követően az így kiszámolt kép segítségével már bármely téglalap alakú terület kiszámítása, tetszőleges skálázás esetén megoldható mindössze néhány tömb-hivatkozás használatával (2. ábra).



2. ábra. Példa az integrált kép felhasználására: a D-vel jelölt területhez tartozó összeg kiszámolása négy tömbreferenciával történik: $4 - 2 - 3 + 1$. [20]

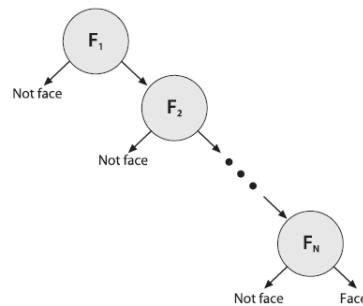
3.2. A betanítás

Az algoritmus alkalmazása során egy kis ablakkal pásztázzuk végig a feldolgozandó képet. Habár az egyes sajátosságok kiszámítása nagyon gyors, egy ilyen ablak méretétől függően nagyon sok szóhajóhető sajátssággal kellene dolgozunk a sajátosság típusából, pozíciójából és skálázásából adódóan (24x24-es ablakméret esetén is több, mint 180 000-el). Mivel ez túlságosan költséges volna, ennél fogva a tanítás során törekszünk arra, hogy csak a legjobban szeparáló sajátosságokat használjuk fel, így jelentősen csökkenteni tudjuk az osztályozó méretét. Lényegében minden újonnan adaptált (gyenge) osztályozó egy az

aktuális eloszlás figyelembevételével a tanítóhalmazt legjobban szeparáló, egyszerű Haar alapú sajátjának feleltethető meg. A módszer hatékonysága természetesen növelhető további sajátjaink felhasználásával, ami viszont a sebesség csökkenésével jár. Freund és Schapire igazolta, hogy tanításkor a hiba mértéke iterációnként exponenciálisan csökken. Emellett igaz, hogy a módszer rövid időn belül, nagyméretű margók kialakítására képes a tanítóhalmaz elemei között és jól általánosít [20].

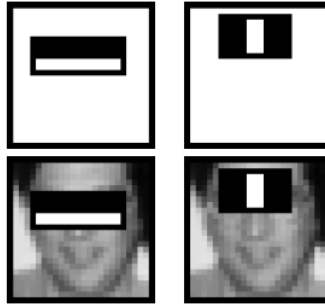
3.3. Kaszkád struktúra

Mivel a fokozatosan növekvő komplexitású osztályozókat egy hierarchikus, kaszkád struktúrába szervezzük (3. ábra), így a kép számunkra közömbös régióival sokkal kevesebbet kell foglalkozunk és a bonyolultabb számításokat már csak a számunkra ígéretesebb területeken kell elvégeznünk. Már mindössze 2 Haar-szerű sajátjának segítségével készíthető olyan arc-detektor, amely a pozitívan besorolandó mintákat az esetek 60, a negatívan besorolandó mintákat pedig az esetek 99%-ában helyesen sorolja be (4. ábra), azaz pusztán ezekkel a sajátjainkkal a felesleges számítások nagy része elkerülhető.



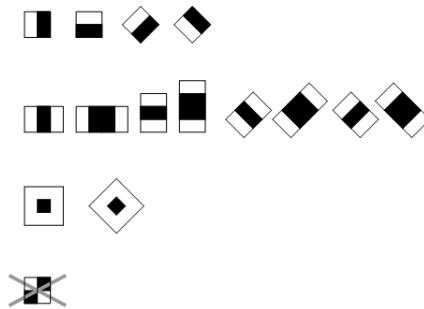
3. ábra. A kaszkád struktúra. [2]

Az aktuálisan vizsgált képrészlet arcképként történő elfogadásához, a kaszkád minden egyes osztályozójának pozitív válasza szükséges. Bármely osztályozójának negatív válasza esetén a régiót elvetjük.



4. ábra. Az AdaBoost által kiválasztott első két sajáttság (felső sor) és ezek illesztése (alsó sor). [20]

Az eddig ismerttetett módszer teljesítményét egy utóoptimalizálási eljárással és a felhasznált sajáttsághalmaz megváltoztatásával javították (5. ábra) [8]. Ennek köszönhetően a hiba a pozitív minták besorolása esetén 10%-kal csökkent.



5. ábra. A sajáttságok új halmaza néhány korábbi 45 fokkal elforgatott változatával és egy jellegében is új sajáttsággal bővült. Az ábrán áthúzott, sakktáblaszerű sajáttságot viszont az új módszer nem használja.

4. Az előfeldolgozás

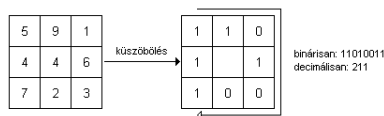
Az előfeldolgozás voltaképp egy bemeneti kép transzformációját jelenti, amely során a jobb felismerési eredmény elérését segítő jellemzők kiemelése történik. Ha a képeket olyan vektorokként képzeljük el, melyek koordinátái a kép pixeleinek felelnek meg, akkor a számunkra legkedvezőbb transzformáció az lenne, amely egyrészt az egy osztályba tartozó képeket az új vektortéren belül egyetlen pontfelhőbe csoportosítaná, másrészt a különböző osztályhoz tartozó pontfelhők közt minél nagyobb távolságot tartana. A mi esetünkben az arckép kivágása után két különböző típusú előfeldolgozást hajtottunk végre. Az előfeldolgozást követi majd az osztályozás.

4.1. Az arc kivágása

A rendszer által használt legegyszerűbb feldolgozás. Az előfeldolgozás a szürkeskálázást, átméretezést és a hisztogram-kiegyenlítést foglalja magában. Paramétere az átméretezéskor használt új méret pixelben. A felhasznált sajátság vektor elemei az így előálló kép pixelértékeinek sorfolytonos felsorolása. A sajátság vektor elemszáma a paraméterben megadott magasság és szélesség szorzata.

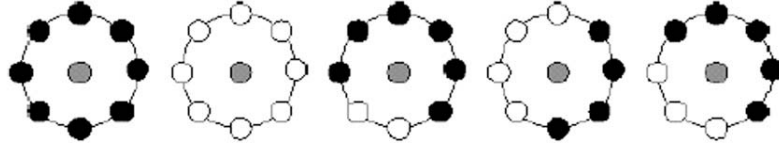
4.2. Local Binary Patterns

Az Ojala és mások által publikált eredeti LBP operátor (6. ábra) a textúra leírás hatékony eszközének bizonyult [10]. Ez az operátor a kép minden pixelét felcímkézi aszerint, hogy az adott, aktuálisan címkézendő pixel 3x3-as szomszédságában mely pixelek rendelkeznek legalább ugyanakkora intenzitással. Ilyen módon az összes lehetséges kombinációhoz rendelhető egy 0 és 255 közötti bináris szám, amit LBP kódnak nevezünk.



6. ábra. Az eredeti LBP operátor.

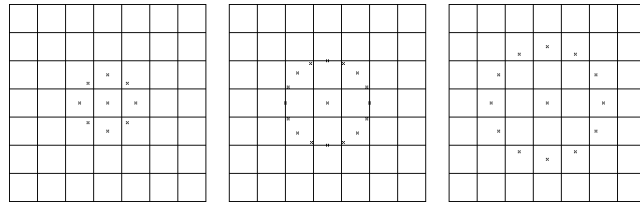
Az így kapott címkék alapján egy, a kép adott régióján számolt hisztogramot használunk a textúra leírására. Ezen LBP kódok mindegyike tulajdonképpen implicite valamely lokális primitívet jelöli: egy (valamilyen módon elhajló) szegélyt, egy sötét foltot, sarokpontot, stb. (7. ábra), tehát ilyen módon minden LBP kód tekinthető egy texton-nak (texton: egy kép vagy videó alapvető mikro-struktúrája).



7. ábra. Minden LBP kód értelmezhető egy lokális primitívként (egyenletes árnyalatú terület, vonalvégződés, sarokpont, él, ...). [16]

Ennek az LBP operátornak nagy gyengesége, hogy csak 8-szomszédság alapján történik az összehasonlítás és így nem alkalmas nagyobb méretű jellemzők vizsgálatára. Az operátort később kiterjesztették, hogy különböző sugarú szomszédságok esetén is alkalmazható legyen [11], bár ez együtt jár a számításigény jelentős megnövekedésével.

A körkörös szomszédságot, és a pixelértékek bilineáris interpolációját feltételezve tetszőleges (akár nem egészértékű) sugár (r), és pontszám (p) használható. A (p, r) jelölés egy p darab pontból álló, r sugarú körön lévő, egymástól egyenlő távolságra elhelyezkedő szomszédsági halmazt jelöl.

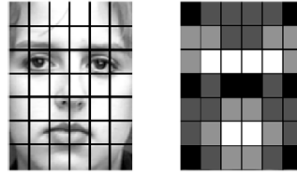


8. ábra. Különböző paraméterű LBP operátorok ($p = 8, r = 1$), ($p = 16, r = 1,5$), ($p = 24, r = 2$).

Egy (p, r) paraméterű LBP operátor - a szóba jöhető bináris minták száma alapján - 2^p darab lehetséges címkeértéket generál. Megmutatható, hogy bizonyos bináris kódok nagyobb információval bírnak, mint egyes más bináris kódok [11]. Ezért bizonyos esetekben elegendő lehet az előbbi 2^p számosságú LBP kódok halmazának egy lényegesen kisebb számosságú részhalmazának használata a textúrák leírásához (ezek az ún. uniform mintázatok [11]). Valamely LBP kódot uniformnak nevezzük, ha azt egy körkörös, bináris sztringként értelmezve legfeljebb két bit átbillenést tartalmaz (pl.: 00001111, 00000000, 10000001). Megmutatták, hogy az uniform mintázatok használatával $(8, 1)$ szomszédság esetén a teljes halmaznak a 87,2%-át, a $(16, 2)$ szomszédság esetében pedig mindössze 70,7%-át szükséges használnunk [11]. Az uniform minták előnye, hogy alkalmazásukkal kisebb elemszámúak lesznek az osztályozáshoz használt sajátosság vektoraink, így az osztályozó állomány mérete csökken és az osztályozás sebessége pedig növekszik.

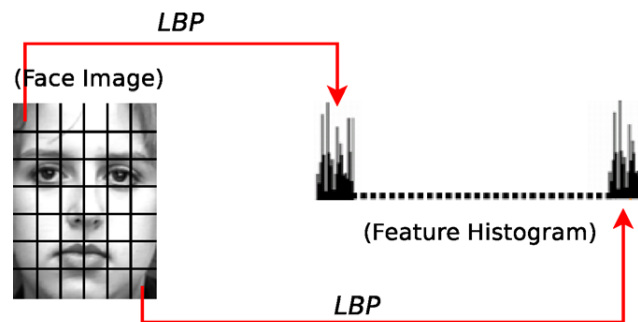
Az eddig ismertetett eljárás elvégezhető akár a teljes arcképen, ezáltal egyetlen hisztogramot generálva, amely mindössze az egyes LBP kódok gyakoriságáról tartalmaz in-

formációt. A gyakorlatban viszont célszerűbb a képet előbb régiókra felbontani, majd ezen régiókhoz külön-külön meghatározni a hozzájuk tartozó hisztogramokat, mert így impliciten már az arc formájáról is adódik információnk.



9. ábra. A súlyozott LBP: a régiók világossága jelzi, hogy az adott terület milyen fontosságú. [16]

Az imént kijelölt régiók eltérő fontosságúak lehetnek egy adott jelenség vizsgálata során (pl.: a szemekhez és a szájhoz tartozó régiók tekinthetők a leghangsúlyosabbaknak). Ennek következtében az egyes régiókhoz súlyt rendelhetünk annak függvényében, hogy mennyire tekintjük őket fontosnak. Ezt a módszert súlyozott LBP-nek nevezzük (9. ábra).



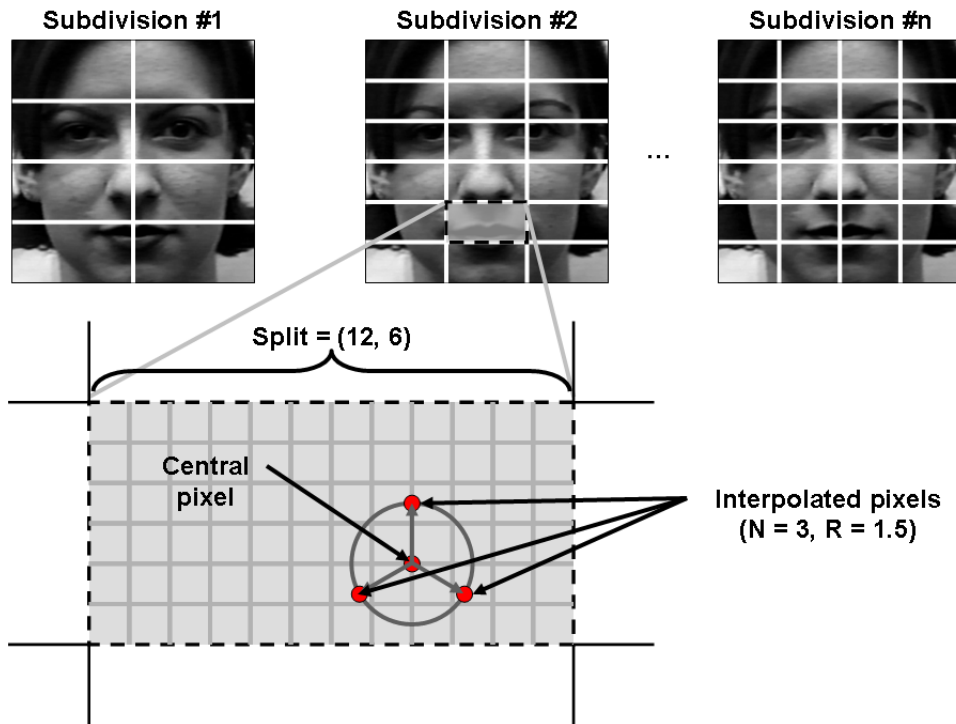
10. ábra. A régiókhoz tartozó hisztogramok konkatenációja. [16]

Az említett hisztogramok konkatenálásával nyerhetjük az osztályozás alapjául szolgáló sajátosság vektort (10. ábra). Megjegyzendő, hogy az általunk használt rendszer (az ún. multisplit LBP (11. ábra)) az előbbi felosztások egy egész sorozatából állítja elő a sajátosság vektort az egyes felosztásokhoz tartozó hisztogramok összefűzése által. A felosztás a következő módon zajlik: definiálunk egy téglalapot, majd a kép középpontjától indulva vízszintes és függőleges irányban is annyiszor illesztjük a képre átfedésmentesen, ahányszor arra ráfér (a kép szélén, az adott felosztásból adódóan bizonyos pixelsávok figyelmen kívül maradhatnak). Ez generálja az adott téglalapsorozat adott téglalapjához tartozó felosztást.

1	2	3		7	8	9		16	17		28	29	30	31	32	33	46	47	48	49	50	51
																	52	53	54	55	56	57
				10	11	12		20	21		34	35	36	37	38	39	58	59	60	61	62	63
4	5	6		13	14	15		22	23		40	41	42	43	44	45	64	65	66	67	68	69
								24	25								70	71	72	73	74	75
								26	27								76	77	78	79	80	81

11. ábra. Példa a kép felosztására multisplit LBP esetén (a számok az adott régióhoz tartozó hisztogram sorszámát jelentik) (a példában nincsenek kimaradt pixelsávok).

A LBP módszer legfőbb előnye a relatíve gyors számolhatóság, továbbá a megvilágítási körülményekkel szemben mutatott invarianciája. Mindemellett stabil marad az alacsony felbontású képek széles tartományában és jól általánosít különböző adatbázisok esetén is. Rendszerünk paraméterei a felosztásokat generáló négyzetek (splits) kiterjedései pixelben, a maszkhoz használt sugár (R), és a maszkban felhasznált pontok száma (N). Sajátság vektor elemszáma: $S \cdot 2^N$, ahol S az összes generálódott felosztáshoz tartozó összes téglalap (tehát a felhasznált hisztogramok) száma.



12. ábra. LBP paraméterek.

4.3. Személyfüggetlen érzelemfelismerés Local Binary Patterns segítségével

Shan C. és Gong S. az LBP segítségével történő érzelem felismerésről írt cikkükben számos osztályozó módszer működését tanulmányozták. Az alábbiakban az LBP mintaillesztéssel,

az LDA-val (Linear Discriminant Analysis) és az LP-vel (Linear Programming) kombinált módszereit ismertetjük.

A mintaillesztés során minden érzelemosztályhoz rendelünk egy etalon képet, ami az adott érzelmet legjobban kifejező tekintetet hivatott ábrázolni. Ezt az egy adott osztályhoz tartozó mintaképek átlagolásával számoljuk ki. Az ehhez tartozó sajátosság vektorral és a legközelebbi-szomszéd módszerrel a további minták már osztályozhatók.

Egy másik, arci érzem tanulására alkalmas technika az LDA. A módszer olyan vetítő tengelyeket keres, amelyekre a mintapontokat vetítvén az egy osztályhoz tartozó adatpontok közel kerülnek egymáshoz, és a különböző osztályokhoz tartozó pontthalmazok pedig minél távolabb. Amennyiben a legnagyobb diszkriminanciát hordozó jellemzőket szeretnénk kiszűrni, az LDA kombinálható PCA-val (Principal Component Analysis).

Az LP esetén egy olyan $f(x)$ lineáris függvényt keresünk, melyre igaz, hogy $f(x) > 0$, ha a beérkező x minta pozitív címkéjű és $f(x) \leq 0$, ha negatív címkéjű.

Shan és Gong az SVM különböző magfüggvények és az eltérő LBP alapú módszerek összehasonlítása során a következő táblázatokban foglalt eredményeket kapták [16]. A 6 osztályú érzelemfelismerés azt jelenti, hogy a természetes arckifejezést nem vették bele az osztályozási kategóriák halmazába.

	6 osztályú érzelemfelismerés	7 osztályú érzelemfelismerés
SVM (lineáris)	91.5 ± 3.1	88.1 ± 3.8
SVM (polinomiális)	91.5 ± 3.1	88.1 ± 3.8
SVM (RBF)	92.6 ± 2.9	88.9 ± 3.5
	6 osztályú érzelemfelismerés	7 osztályú érzelemfelismerés
SVM (lineáris)	87.7 ± 4.7	80.2 ± 4.9
LDA + NN	79.2 ± 7.2	73.4 ± 5.6
	6 osztályú érzelemfelismerés	7 osztályú érzelemfelismerés
SVM (lineáris)	90.4 ± 3.9	86.0 ± 3.3
LP	89.6 ± 3.6	82.3 ± 3.5

1. táblázat. Eredmények. [16]

4.4. Érzelemfelismerés boosting Local Binary Patterns-el

Az eddigiekben csak rögzített felosztással vagy felosztássorozattal dolgoztunk. A probléma, hogy ekkor a származtatott sajátosságok nagyban függni fognak a felosztásban megjelenő régiók helyzetétől és méretétől. Boosting során célunk, hogy megtaláljuk az adott számú legfontosabb, a megkülönböztetést leginkább segítő tartományt (ezek eltérő kiterjedésűek és helyzetűek lehetnek). A jobb eredmény elérése érdekében csak ezeket fogjuk használni.

Shan és Gong az AdaBoost-ot SVM-mel, illetve az előbbi alfejezetben bemutatott mintaillesztő és LDA módszerekkel kombinálva vizsgálta [16]. Ennek eredményeként mondhatjuk, hogy az érzelem felismerés esetében a keresett területek jellegzetesen a szem és a száj környékén találhatók.

4.5. A biológiai nem felsimerése boosting Local Binary Patterns-el

Sun és mások [17]-ban két AdaBoost-on alapuló kísérletet végeztek a nem felismerésére. Az első esetben a bejövő, kivágott arcképeket egyenlő méretű régiókra osztják a másik esetben pedig AdaBoost segítségével próbálják megtalálni a diszkriminatív szektorokat.

Mindkét esetben a FERET adatbázisból összeválogatott 2000 képből álló tanító és 400 képből álló teszhalmaz segítségével történt a vizsgálat. A képek 256 színmélységűek voltak melyen előzetesen 144x120-as mérete skálázást, hisztogram kiegyenlítést végeztek. A tanítóhalmazban 1200 férfi és 800 nő szerepelt, a teszhalmazban pedig egyenlő arányban 200-200 képet használtak.

Az első kísérlet során a képeket 36 db megegyező (24x20-as) méretű régióra osztották fel. A Self Organizing Maps módszer segítségével a tanítóhalmazt 10 részre szeparálták. A kísérlet eredményeképpen azt tapasztalták, hogy a leginkább eltérő régiók a férfiak és nők között a szemöldök, orrnagyereg és az áll környékén találhatók. Az osztályozás során a súlyozott Khí négyzet távolságot alkalmazták:

$$x_w^2(S, M) = \sum_{i,j} w_j \frac{(S_{i,j} + M_{i,j})^2}{S_{i,j} + M_{i,j}},$$

ahol S és M két LBP hisztogram és w_j a j -ik régióhoz tartozó súly. Ezt követően a beérkező mintát a legközelebbi, a megfelelő osztályhoz tartozó mintához illesztették.

A második kísérlet esetében, az AdaBoost eltérő méretű és helyzetű ablakok közül választja ki a megfelelő sajátságokat. Az eredményül kapott három legfontosabb sajátságot a 13. ábrán láthatjuk.



13. ábra. Az AdaBoost által kiválasztott első három legfontosabb régió. [17]

A teljesítmény az első esetben 82.75%, a másodikban 95.75% volt. Ezek alapján megállapítható, hogy az LBP operátor hasznos a nemek megkülönböztetése során, sőt hasznosabbnak bizonyult, mint az arcdetektálás során alkalmazott Haar-szerű sajátságok [17]. Továbbá kiderült, hogy a skálázható és mozgatható ablakok segítségével jobb eredményt érhetünk el.

5. Számítógépes tanulás

A számítógépes tanulás során empirikus (tapasztalati) adatok segítségével próbáljuk egy algoritmus működésén javítani. Sok típusát megkülönböztetjük és minden típusához megtalálhatók a rendelkezésre álló módszerek. Az adatok lehetnek ellenőrzött (címkézett) adatok, vagy ellenőrizetlen (címkézetlen) adatok. Minden esetben felcímkézett adathalmazzt használunk, ahol adat alatt egy képet és címke alatt a hozzá tartozó, megfelelő annotációt értjük.

Tapasztalatom szerint az osztályozók (AdaBoost, SVM) nem adtak nagy különbséget, hanem inkább az előfeldolgozás számít. Ennek következményeként a betanításra SVM-et használtam.

5.1. AdaBoost

Az AdaBoost tanulás egy egyszerű, de hatásos módja egy nemlineáris függvény megtanulásának. Az AdaBoost a betanítás során meghatároz néhány gyenge osztályozót (amelyektől csak azt várjuk el, hogy a véletlenszerű találgatásnál jobb eredményt szolgáltatassanak), és ezek együttesét lépésenként, egy erős és hatékony osztályozóvá erősíti.

Feltételezzük a tanító minták egy kezdeti eloszlását. Minden iteráció során kiválasztunk egy olyan osztályozót, amely az aktuális eloszlást figyelembe véve minimalizálja a súlyozott hibarátlát. Ezután módosítjuk az eloszlást: a hibásan osztályozott minták súlyát növeljük, a többi minta súlyát pedig csökkentjük. Ennek hatására a továbbiakban kiválasztandó gyenge osztályozóink majd jobban fókuszálnak a rosszul besorolt mintákra.

Az AdaBoost egy széles körben elterjedt eljárás. Például Schapire és Singer [14] bemutattott egy boosting alapú rendszert szövegkategorizálásra, továbbá Yang és mások [26] javasoltak egy Gabor sajátságokat használó AdaBoost eljárást arc felismerésre. Mindenesetre legfontosabb alkalmazása minden bizonnyal az arcdetektálás. (Ezen a ponton jegyezném meg, hogy az arcdetektálás és arcfelismerés eltérő fogalmak. Az arcdetektálás során egy képen keresünk arcokat, hogy azokat további feldolgozásnak vessük alá. Az arcfelismerés során viszont egy arcképen szereplő személy személyazonosságát meghatározzuk meg.)

5.2. Support Vector Machine

Az SVM egy ellenőrzött tanulást segítő, regressziós és osztályozási problémák esetén alkalmazható módszer (tehát olyan szituációkban, amikor minden tanítómintához ismerjük a hozzá tartozó osztálycímkeit). Az arckifejezések felismerésére is hatékonyan működő technika. Az SVM voltaképpen egy olyan speciális neurális hálózat, amely az adatokat egy, az

adatok eredeti vektorterénél magasabb (akár végtelen) dimenziószámú terébe képezi le, majd a tanítómintákat lineárisan szeparáló, maximális margójú hipersíkot szolgáltatja.

Ha adott a tanítóadatok egy $\{(x_i, y_i), i = 1, \dots, j\}$ felcímkézett halmaza, ahol $x_i \in R^n$ és $y_i \in \{1, -1\}$, akkor egy új x teszt mintát az alábbi módon osztályozunk:

$$f(x) = \text{sgn}\left(\sum_{i=1}^j \omega_i y_i K(x_i, x) + b\right) \quad (i = 1, \dots, j),$$

ahol a ω_i -ik egy duális optimalizációs feladat Lagrange multiplikátorai amely a szeparáló hipersíkot írja le, a $K(\cdot, \cdot)$ egy kernel függvény, és b a hipersík küszöbölő paramétere. A pozitív ω_i -ikhoz tartozó x_i vektorokat nevezzük tartóvektoroknak (support vector). Az SVM a tanulás során ezen tartóvektorok és a hipersík közötti távolságot maximalizálja. A leggyakrabban előforduló kernel függvény típusok: lineáris, polinomiális, RBF (Radial Basis Function).

Az SVM bináris döntéseken hoz, ennek folytán az ennél összetettebb (kettőnél több címkéjű) osztályozásokat is bináris esetekre vezetjük vissza. Például egy konkrét (pl.: a boldogságot vizsgáló) osztályozó csak annyit hivatott megállapítani, hogy a vizsgált minta az adott osztályba sorolandó, vagy valamelyik (bármelyik) másikba. Vagy másképp mondva: egy adott szempont (pl.: arcifejezés) vizsgálata során, annak minden egyes osztályához (pl.: szomorúság, félelem, undor...) külön-külön betanításra van szükség.

Ezek az osztályozók szintén kaszkádba szervezhetőek. Például először megvizsgálható, hogy egy arcifejezés természetes-e. Amennyiben nem az, akkor tovább vizsgáljuk mondjuk az öröm szerinti osztályozóval. Hogyha ismét negatív az osztályozónk válasza, megnevezhetjük az undor szerint. Ez az elgondolás kutatásaim egyik további ága lehet.

5.3. Incremental Support Vector Machine

Az előbb ismertetett módszer alkalmazásához bemeneti adatként mindig rendelkezésre kell állnia a teljes tanítóhalmaznak, ami nem feltétlenül praktikus a gyakorlatban (pl.: gondoljunk hatalmas méretű tanítóhalmazra, ilyenek tárolása meglehetősen költséges lehet). Ráadásul ez a teljes tanítóhalmaz utólag azonban nem is feltétlenül érhető el, ugyanis az osztályozás végrehajtásához számunkra elegendő mindössze a tanulás során alkalmazott, a tanulásra valóban hatással levő tartóvektorokat (azaz csak bizonyos bemeneti minták halmazát) megtartani, ami általában csak egy valódi részhalmaza a kiinduló halmaznak. Ez a részhalmaz új betanítás elvégzésére önmagában, bizonyos más információk jelenléte nélkül azonban nem elegendő, mert csak közelítő eredményt szolgáltat [18]. Ez a "közelítő eredmény" a mi esetünkben teljesen használhatatlan osztályozókat eredményezett a gyako-

ratban.

Ezzel szemben lehetséges az osztályozás utólagos, hatékony javítása új mintaelemek hozzáadásával (vagy kiiktatásával) az ún. online tanulás módszerével, amely a tartóvektorok halmazának tárolása mellett bizonyos járulékos információkat is nyilvántart. Az online tanulás során mindössze egyetlen új, címkézett tanítómintát illesztünk a csupán implicit módon rendelkezésre álló tanítóhalmazhoz. Egy megvalósításról olvashatunk [5]-ben.

6. A rendszer részletes bemutatása

6.1. A rendszer felépítése

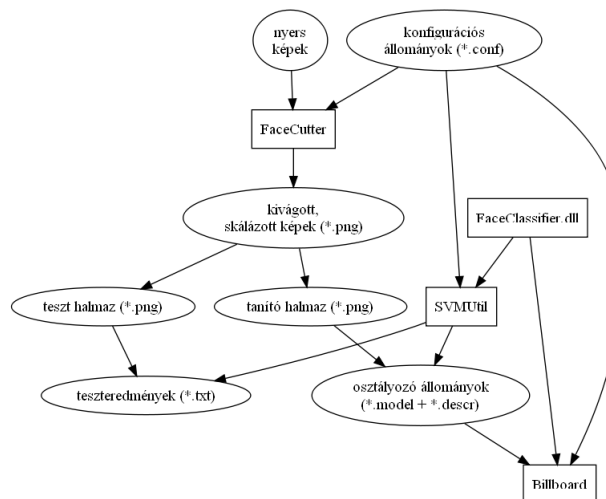
A rendszer három alkalmazásból, a hozzájuk tartozó konfigurációs állományokból és egy dinamikus könyvtárból épül fel. A rendszerkomponensek egymáshoz való viszonya a 14. ábrán látható.

FaceClassifier dinamikus könyvtár: Az alkalmazások által használt betanító, tesztelő, osztályozó függvényeket tartalmazza. Minden más alkalmazás ezen dinamikus könyvtár eszközeit használja.

FaceCutter alkalmazás: A betanításhoz használt képek előkészítéséhez készített parancssoros segéd-alkalmazás. A bemeneti képen megtalált arc kivágásra kerül, majd a facecutter.conf konfigurációs állomány alapján szűrkeskálázza, átméretezi.

SVMUtil alkalmazás: Az SVM betanítása során alkalmazott parancssoros segédeszköz. Az előkészített képek és az adott konfiguráció alapján elkészíti az osztályozó állományokat (*.model) és a hozzájuk tartozó leíró állományokat (*.descr).

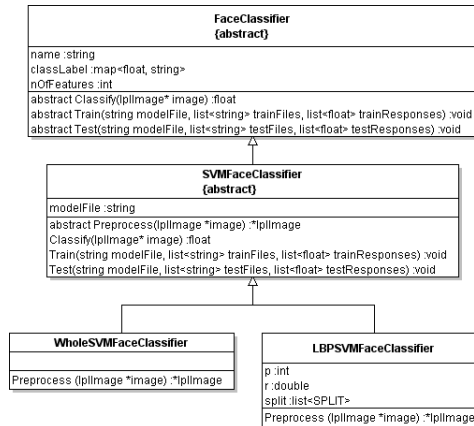
Billboard alkalmazás: A reklámot megjelenítő alkalmazás. A reklámok a billboard.conf konfigurációs állomány alapján rendelődnek az egyes osztályokhoz.



14. ábra. A rendszer felépítésének vázlata.

6.2. A FaceClassifier dinamikus könyvtár

A segéd- és a fő alkalmazás által használt könyvtár. A főbb osztályok lényegesebb attribútumai és metódusai a 15. ábrán láthatóak.



15. ábra. A dinamikus könyvtár vázlat szerű UML diagramja.

A rendszer moduláris felépítéséből adódóan könnyen bővíthető új osztályozókkal. Ehhez a `Preprocess(IplImage *image)` függvényt kell úgy implementálni, hogy az, az argumentként kapott kép egy másolatán elvégezze az adott osztályozóhoz tartozó előfeldolgozási lépéseket.

6.3. A konfigurációs állományok

Az alkalmazásokhoz tartozó konfigurációs állományok XML formátumúak. Az ezekben lévő bejegyzések alapján történik a képkivágás, a betanítás, a tesztelés és a reklámképek besorolása. Az alkalmazások által használt bejegyzéseket az alábbi táblázatok foglalják össze.

A Facecutter konfigurációs bejegyzései	
config/haarcascade	Az arc detektáláshoz használt cascade fájl elérési útvonala.
config/image/grayscale	Itt állítható be, hogy a betanításra előkészítendő képeket szükséges-e szürkeskálássá alakítani. Lehetséges értékek: yes / no.
config/image/{width, height}	Az új, átméretezett kép szélessége és magassága (width = height = 0 esetén nem történik átméretezés).

Az SVMUtil konfigurációs bejegyzései	
config/classifierType	A SVM betanítása során használt osztályozó típusa. Lehetséges értékei: WholeSVMFaceClassifier / LBPSVMFaceClassifier.
config/svmparameters/{kernelType, degree, gamma, coef0}	Az SVM betanításának paraméterei.
config/image/{width, height}	Az új, átméretezett kép szélessége és magassága (width = height = 0 esetén nem történik átméretezés).

InteractiveBillboard konfigurációs bejegyzései	
config/haarcascade	Az arcdetektáláshoz használt cascade fájl elérési útvonala.
config/input	Amennyiben nincs camerastream, akkor az itt megadott videófájlt használja a program (opcionális).
config/classifiers/	A billboard osztályozóinak felsorolása.
/name	Az osztályozó neve.
/modelfile	Az osztályozó fájl elérési útvonala.
/descrfile	Az osztályozóhoz tartozó leíró állomány elérési útvonala.
/neg	Az osztályozó negatív válaszának szöveges interpretációja.
/pos	Az osztályozó pozíció válaszának szöveges interpretációja.
config/ads/ad/	A reklám attribútumainak felsorolása.
/image	A feltételek teljesülésekor megjelenítendő hirdetéshez tartozó kép elérési útvonala.
/conditions/negresponses	Azon osztályozók neveinek a felsorolása, amely negatív választ kell, hogy adjanak az adott hirdetés esetén.
/conditions/posresponses	Azon osztályozók neveinek a felsorolása, amely pozitív választ kell, hogy adjanak az adott hirdetés esetén.

6.4. A vibrálás kiküszöbölése

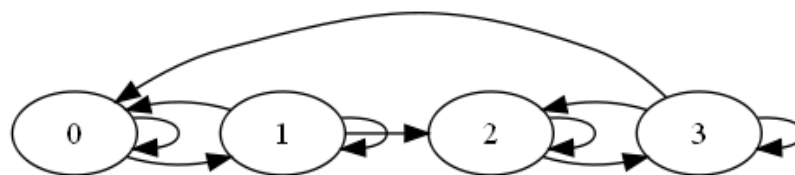
Mindeddig az elsődleges törekvésünk az volt, hogy minél pontosabb és minél megbízhatóbb osztályozókat készítsünk. Viszont mivel egy osztályozás végeredményét sosem tekinthetjük

teljesen biztosnak, így előfordulhat az az eset (és a gyakorlatban ez természetesen nem ritka), hogy egy, a kamera előtt tartózkodó személyt a rendszer (a pillanatnyi belső reprezentáció függvényében) eltérő osztályokba sorol (ehhez akár a fényviszonyok átmeneti megváltozása is elegendő lehet). Mindemellett a helyzetet rontja, hogy sajnos az is elképzelhető, hogy az arcdetektorunk téved és egy arcképet átmenetileg nem talál meg. Ez a reklámok esetleges gyors, vibrálásszerű váltakozásával járna, de mivel ez nagyon zavaró lenne, ezért ezt ki kell küszöbölnünk egy állapotgép implementációjával.

Az alapgondolat az, hogy csak akkor jelenítsünk meg reklámot, hogyha már szinte biztosak vagyunk abban, hogy valaki valóban áll a kamera előtt: azt mondhatjuk, hogy egy személy tartózkodik a kamera előtt, amennyiben azt az arcdetektornak bizonyos számú frame-n keresztül folyamatosan sikerül detektálnia (tehát ezen „időtartam” alatt, minden egyes frame-n legalább egy arcképet detektáltunk). Ehhez hasonlóan a személy kamera elöl történő elmozdulását, eltűnését sem tekintjük rögtön biztosnak.

Az 16. ábra mutatja az egyes állapotok közti elmozdulási lehetőségeket. Az állapotok intuitíve a következő eseteket jelenti:

- 0. állapot: A kiinduló állapot. Feltételezzük, hogy nem tartózkodik senki a kamera előtt. Az 1. állapotba ugrunk, amennyiben sikeres arcdetektálás történik.
- 1. állapot: A meglévő szempontok szerint osztályozzuk a frame-n található arcot. Egy változót inkrementálunk, amíg el nem ér egy meghatározott küszöbszámot. Ekkor a 2. állapotba ugrunk. Amennyiben még ez előtt sikertelen arcdetektálás történik, visszakerülünk a 0. állapotba.
- 2. állapot: Mostmár biztosnak tekintjük, hogy valaki figyeli a hirdetőtáblát. Ebben az állapotban megjelenítjük az 1. állapotban történt osztályozás eredményéhez tartozó reklámot. Amennyiben nem találunk arcot a videostream-en, a 3. állapotba jutunk.
- 3. állapot: Innen sikeres arcdetektálás esetén visszakerülünk 2.-be, viszont meghatározott számú, egymást követő, sikertelen detektálás után a személy eltűnését feltételezzük és ismét a 0. állapotba kerülünk.



16. ábra. A vibrálás kiküszöbölésére alkalmazott állapotgép.

6.5. A tanuláshoz és a tesztekhez kialakított adatbázisok

Az osztályozókat két arcképadatbázis felhasználásával alakítottuk ki. Ezek a Cohn-Kanade: AU-Coded Facial Expression Database és a FERET adatbázisok voltak. Az érzelmek vizsgálatára a Cohn-Kanade, az életkor-, és a bőrszín meghatározására a FERET, a nem esetében pedig mindkét adatbázist alkalmaztuk. A FERET adatbázis esetében nagy segítséget nyújtottak az életkor becslésére, a nemre, és az embercsoportra vonatkozó leíró állományok, amelyek az egyes képekhez tartozó címkéket tartalmazták. A Cohn-Kanade adatbázis esetében az előbbihez hasonló állományok hiánya miatt az arcképeket manuálisan annotáltam.

A FERET alapú adatbázisok hatékony kialakításához szükség volt az adatbázis egyfajta „előfeldolgozására”:

- a képeket a leíró állományok alapján, nem, életkor és embercsoport szerint partícionáltam,
- eltávolítottam a szemüveges és a nem profil beállítású képeket,
- ezután minden személyről legfeljebb egy képet tartottam meg, a személyhez tartozó többi kép törölve lett.

Az egyes osztályozásokhoz tartozó adatbázisok kialakítása a szükséges fájlok kiválogatásával majd a megfelelő könyvtárstruktúrába való összemásolásával történt, relatíve nagy méretű könyvtárakat eredményezve. Megjegyzem, hogy ez a módszer azonban nem mutatkozott hatékonynak, ugyanis ilyen sok állományt tartalmazó, nagy méretű könyvtárak mozgatása, tárolása körülményes volt (a rengeteg duplikációról nem is beszélve). Ezért a továbbiakban célszerűen a szükséges fájlok helyett csak a különböző kategóriákhoz tartozó fájlok nevei kerülnek majd összegyűjtésre (egy-egy külön állományban), majd ezt a fájlnév gyűjteményt felhasználva, egy segédalkalmazás (vagy szkriptek) segítségével fogjuk a tanulást / tesztelést előkészíteni a merevlemezen, az éppen szükséges képek megfelelő helyre történő másolásával. Lényegében azt mondhatjuk, hogy a fizikai összegyűjtés helyett logikai csoportosítást végzünk.

6.6. Az egyes adatbázisok

- Az érzelemfelismeréshez tartozó képadatbázis: Hét különböző érzelmi osztályt különböztetünk meg: természetes, vidám, szomorú, haragos, megrémült, undorodó és csodálkozó. Az egyes képek besorolása alapján a fenti kategóriák valamelyikébe csoportosítottuk őket, majd a képek tananyag (a betanításhoz használt) és tesztanyag

(a teszteléshez használt) halmazokra történő felosztása történt. Az első teszt esetében kevés képen (4-7 db), a második teszt esetében pedig a teljes Cohn-Kanade adatbázison vizsgáltuk az eredményt.

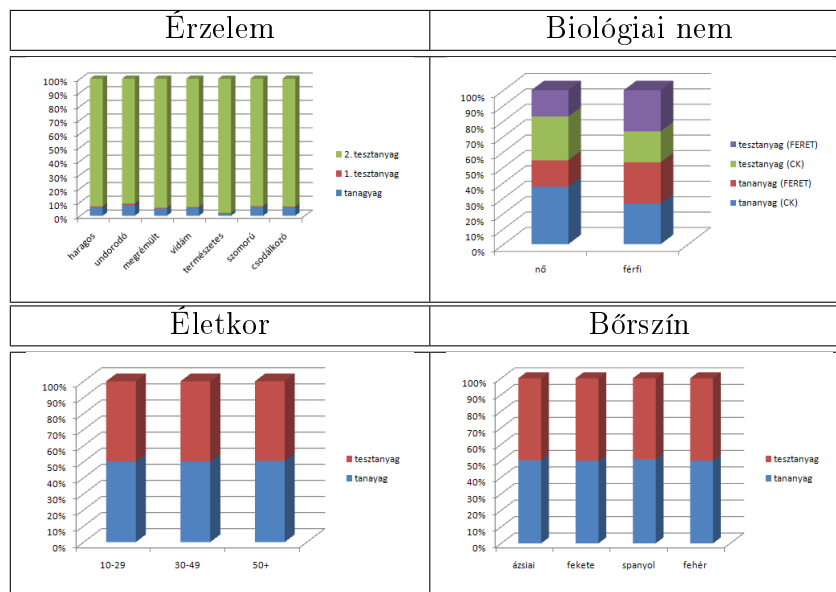
- A biológiai nem felismeréséhez tartozó Cohn-Kanade alapú képadatbázis: A két osztály: nő, férfi. Minden személyhez, minden (rendelkezésre álló) érzelmi kategóriából 3 kép tartozik (így növelve a hatékonyságot az érzelmi állapotok kiváltódása esetén). A személyek egyik felét használtuk a betanításhoz, a másik felét pedig a teszteléshez.
- Az életkor meghatározásához tartozó képadatbázis: Az életkor esetében a 10-29, a 30-49 és 50+ osztályokat különböztetjük meg (a FERET adatbázis nem tartalmaz képet 10 évnél fiatalabb személyről).
- A bőrszín meghatározásához tartozó képadatbázis: A FERET szemléletétől valame-lyest eltérően – a felhasználható képek számából adódóan - következő osztályok tűntek praktikusnak: fekete, fehér, spanyol, ázsiai.
- A biológiai nem felismeréséhez tartozó FERET alapú képadatbázis: Minden szemé-lyről egyetlen (semleges érzelmi állapotú) képet használtunk fel (a Cohn-Kanade alapú adatbázistól eltérően). A teszt itt is az alanyok felére vonatkozik.

Az adatbázisokat szemlélteti a következő táblázat:

Szempont	Partíciók	Alap
Nem	{nő, férfi}	Cohn-Kanade
Nem	{nő, férfi}	FERET
Érzelem	{vidám, haragos, megrémült, természetes, undorodó, csodálkozó}	Cohn-Kanade
Életkor	{10 - 29, 30 - 49, 50+}	FERET
Bőrszín	{fekete, fehér, spanyol, ázsiai}	FERET

2. táblázat. Az adatbázisok.

A tananyag és a tesztanyag mennyiségi eloszlását a következő diagramok szemléltetik:



3. táblázat. A tananyag és a tesztanyagok mennyiségi megoszlása a különböző osztályozók esetén.

6.7. Eredmények

A mérésekhez egy arc kivágó és 8 eltérő LBP konfigurációt alkalmaztunk. A különböző paraméterezésű osztályozók által szolgáltatott eredményeket az alábbi táblázatokban gyűjtöttük össze. A táblázatokban a legjobban osztályozó konfigurációkat félkövérrel szedett számokkal jelöljük.

Súlyozással megoldható, hogy a pozitív, és a negatív halmazok közti elemszámból eredő különbség ne játsszon szerepet (pozitív halmaz: a betanítás avagy a tesztelés során használt, pozitív címkéjű minták halmaza; negatív halmaz: a betanítás avagy a tesztelés során használt, negatív címkéjű minták halmaza). A táblázatok a pozitív és a negatív halmazon belül sikeresen felismert minták százalékos arányainak átlagait tartalmazzák. Erre a normálásra azért került sor, mert lehetnek olyan szituációk, amikor a pozitív és a negatív halmaz számossága nagyságrendben különbözhet. Mi viszont azt szeretnénk, hogy egy az osztályozás során vétett hiba mindkét halmazon belül, az adott halmaz elemszámával arányos módon érvényesüljön. Más szóval, ha például a betanítás során a pozitív halmazunk számossága lényegesen kisebb, mint a negatív halmazunk számossága, akkor egy a pozitív halmazbeli hibás számítást relatíve súlyosabb hibaként kell értékelnünk. Hogyha nem vesszük figyelembe a halmazok elemszámát, az megtévesztő eredményt szolgáltat, ugyanis képzeljük el a következő szituációt: a teszteléshez egy 10 elemű pozitív halmazt és egy 90 elemű negatív halmazt használunk. Most tegyük fel, hogy korábban, a betanítás eredményeként létrejött osztályozó minden tesztkép esetében negatív választ

ad. Így azt mondhatnánk, hogy a teszt 90%-ban sikeres, holott egyetlen pozitív képet sem sikerült megkülönböztetnünk a negatív képektől.

Az érzelmek felismerése

A kevés képen végrehajtott gyors-teszt eredményei:

	Kivágás	LBP 1	LBP 2	LBP 3	LBP 4	LBP 5	LBP 6	LBP 7	LBP 8
Harag	37.27	69.69	46.66	48.48	51.81	57.27	56.96	39.39	47.57
Undor	77.20	77.20	55.88	81.61	81.61	75.00	71.32	72.05	76.47
Félelem	47.05	50.73	58.08	60.29	58.08	50.00	69.11	56.61	46.32
Öröm	91.24	98.38	91.24	92.85	100.00	98.38	96.77	98.38	96.77
Természetes	59.44	85.71	64.28	66.58	66.58	68.20	69.81	64.97	69.81
Szomorúság	68.48	56.06	33.33	42.42	50.00	54.54	51.51	43.93	50.00
Csodálkozás	90.10	91.66	91.66	83.33	83.33	100.00	91.66	83.33	91.66
Átlag	67.25	75.63	63.01	67.93	70.20	71.91	72.44	65.52	68.37

Table 4: Az első érzelem teszt.

A teljes Cohn-Kanade adatbázison végrehajtott teszt:

	Kivágás	LBP 1	LBP 2	LBP 3	LBP 4	LBP 5	LBP 6	LBP 7	LBP 8
Harag	60.40	59.23	58.74	52.82	55.44	67.48	60.06	58.28	52.66
Undor	75.70	73.36	51.60	76.60	67.25	61.45	73.56	72.94	60.69
Félelem	54.42	50.80	56.42	54.92	52.57	54.90	71.79	51.50	48.87
Öröm	91.18	92.21	87.63	89.90	92.28	91.91	92.66	91.52	92.16
Természetes	77.41	76.30	66.84	70.12	75.11	73.10	76.98	75.74	75.06
Szomorúság	87.06	52.06	60.01	71.80	50.00	52.35	51.40	61.31	54.71
Csodálkozás	84.10	83.55	78.32	79.57	82.63	81.18	84.55	83.48	82.31
Átlag	75.75	69.64	65.65	70.81	67.89	68.91	73.00	70.68	66.63

Table 5: A második érzelem teszt.

Az érzelem felismerésével kapcsolatban az alábbi következtetéseket vonjuk le:

- a harag, az undor és a félelem a többi érzelmenél nehezebben ismerhető fel,
- a nagyobb sugarú LBP operátorok (LBP 3, 6) hatékonyabbak,
- az arckép nagyobb részekre darabolása (LBP 4, 5, 6) jobb eredményt hoz,
- bizonyos esetekben elegendő lehet az arckép egyszerű kivágása

A biológiai nem felismerése

A nem felismerésére kettő (egy Cohn-Kanade (CK) és egy FERET alapú) osztályozót hoztunk létre, melyekkel összesen négy tesztet hajtottunk végre:

- egyet-egyet a betanítás alapjául szolgáló saját adatbázison,
- további egyet-egyet a másik adatbázison, ezzel vizsgálván az osztályozók adatbázisok közötti általánosító képességét.

	Kivágás	LBP 1	LBP 2	LBP 3	LBP 4	LBP 5	LBP 6	LBP 7	LBP 8
Teszt 1	84.77	82.70	77.59	81.99	77.77	76.86	83.64	78.23	81.83
Teszt 2	87.11	79.50	71.82	77.57	77.62	78.99	81.47	78.96	80.66
Teszt 3	74.11	73.81	77.85	76.40	79.36	75.52	71.02	76.80	80.16
Teszt 4	52.98	59.80	65.71	65.00	67.70	56.07	61.77	62.25	62.69
Átlag	74.74	73.95	73.24	75.24	75.61	71.86	74.47	74.06	76.33

Table 6: A biológiai nem felismerése.

Mindkét osztályozó esetében megfigyelhető, hogy saját adatbázison szinte minden konfiguráció esetén jobb eredmények adódtak. Figyelemre méltó, hogy a Cohn-Kanade alapú osztályozó mindkét adatbázison közel azonos eredményt szolgáltatott. Ezen túl, mivel a nem felismerés esetében leginkább a fej formája számít, így az arc kivágása már önmagában egy viszonylag jónak mondható osztályozást generál.

Az életkor felismerése

	Kivágás	LBP 1	LBP 2	LBP 3	LBP 4	LBP 5	LBP 6	LBP 7	LBP 8
10-29	62.42	63.79	63.72	63.52	66.32	65.03	63.06	64.04	66.41
30-49	59.88	54.31	53.81	56.51	58.72	59.51	56.49	59.71	56.56
50+	58.40	76.62	50.00	50.00	75.44	75.42	71.43	72.79	75.02
Átlag	60.23	64.90	55.84	56.67	66.82	66.65	63.66	65.51	65.99

7. táblázat. Az életkor tesztje.

Mivel a fenti táblázat tükrében elmondhatjuk, hogy az életkorral kapcsolatban a rendszer túlságosan gyakran téved, így ez esetünkben majd további erőfeszítéseket kíván. Jelen pillanatban a 10-29 életkorú és az 50 évesnél idősebb korosztály felismerése történik a legmegbízhatóbban.

A bőrszín felismerése

	Kivágás	LBP 1	LBP 2	LBP 3	LBP 4	LBP 5	LBP 6	LBP 7	LBP 8
Ázsiai	67.07	67.68	56.70	54.57	70.88	68.59	69.20	71.34	65.54
Fekete	72.57	54.05	50.00	50.00	59.05	51.35	51.35	65.94	50.00
Spanyol	50.00	50.00	50.00	50.00	52.00	50.00	50.00	49.87	50.00
Fehér	76.43	75.49	64.87	63.13	74.84	76.04	76.92	77.13	74.20
Átlag	66.51	61.80	55.39	54.42	64.19	61.49	61.86	66.07	59.93

Table 8: A bőrszín felismerése.

A fenti táblázatban szembetűnő a viszonylag nagy számban előforduló 50%-os eredmény. Esetünkben ennek több kiváltó oka lehet: egyrészt elképzelhető, hogy az osztályozási tévedések a megvilágítási körülményekből adódnak, másrészt lehetséges, hogy egy – valószínűleg a tanítóhalmaz kis elemszámából kifolyólag - nem megfelelő betanítás eredményeként létrejött, gyakorlatilag konstans válaszunak tekinthető (valójában teljesen rosszul működő) osztályozót jelent, ahol a beérkező minták között nem tudunk különbséget tenni. Ennek következményeként a spanyol bőrszín szerinti osztályozót nem tekinthetjük a gyakorlatban alkalmazható eszköznek.

6.8. Konzekvencia

A teszteket egybevetvén mondhatjuk, hogy a nagyobb maszkok (LBP 4, 5, 6) az érzelem vizsgálata esetén gyakran, más esetben csak ritkán vezettek jobb eredményre. A felosztások sorozatát használó multisplit LBP 7 a bőrszín osztályozásánál relatíve hatékonynak mutatkozott, azonban levonhatjuk a következtetést, miszerint a multisplit LBP (LBP 7, 8) használata nem hozott meggyőző eredményeket. Ezentúl megállapíthatjuk, hogy a szürkeskálás képek esetében nincs elegendő színinformációnk az életkor és az etnikai csoportok felismeréséhez.

7. A rendszer működés közben

A rendszer által használt reklámokat az alábbi módon rendeltük a megfigyelő életkorához és neméhez:

életkor / nem	nő	férfi
10 - 29	kozmetikumok	számítógép, hifi rendszer
30 - 49	ékszer, szakácskönyv	autó
50 +	hajfesték, ránctalanító	-
?	turizmus	mobiltelefon

9. táblázat. A reklámok.

Az alábbiak az alkalmazást szemléltetik működés közben:



10. táblázat. Az alkalmazás működés közben.

Az eddigi eredmények persze mindössze illusztráció jellegűek, mivel még számos más szempont figyelembe vehetnénk, olyanokat például, hogy szemüveges-e az illető, visel-e ékszer, milyen érzelmi állapotban van, stb. Vagy például a reklámot megfigyelőkről –

a hatékonyságot növelendő - készíthetnénk statisztikákat, hogy ki, mennyi ideig nézte, hogyan hatott rá a reklám és így tovább.

8. Összefoglalás

A dolgozatomban igyekeztem bemutatni, hogy egy emberi arc számítógéppel történő vizsgálata milyen megoldandó problémákat vet fel, továbbá, hogy alkalmanként milyen nehézkes lehet ezeknek a problémáknak a hatékony megoldása. Ezen felül minden bizonytalansággal érzékelhető volt, hogy az időről-időre szellemesebb, fejlettebb és jobb hatékonyságot ígérő ötletek, illetve megvalósítások ellenére egy olyan tudományágról van szó, melynél gyakran be kell érniük olyan technikákkal vagy módszerekkel, amelyek eredményeit sohasem tekinthetjük a végtelenségig biztosnak. Elég csak arra gondolnunk, hogy egy személy külső jellemzőinek meghatározása gyakorta önmagunknak is problémát jelenthet, így nem várhatjuk, hogy az emberi agy működését algoritmikusan közelíteni próbáló számítógépekre 100%-osan rábízhatunk ehhez hasonló problémákat.

Megemlítem, hogy a projekt kapcsán készítettem egy olyan online alkalmazást, melynek segítségével bizonyos személyeket annotáltunk életkor, bőr-, haj- és szemszín alapján. Ezek a problémák ugyan nem mind kapcsolódnak közvetlenül az imént tárgyalt feladatokhoz, azonban talán jól érezhető, hogy az ember által szubjektív módon érzékelt jelenségek tanulmányozása során mennyire körültekintőnek kell lennünk.

Továbbá fontos kihangsúlyozni, hogy ugyan az informatika egy intenzíven kutatott, gazdag ismeretanyaggal rendelkező érdekes és ígéretes területéről beszélünk, azonban az eltérő körülmények között végrehajtott, különböző tanulmányok által szolgáltatott következtetések összevetése nem egyszerű feladat.

Munkám során azt tapasztaltam, hogy a dolgozatomban bemutatott rendszer az érzelem és a biológiai nemek felismerésének vonatkozásaiban jól működőnek mondható. Az életkor és a különböző rasszok megbízható besorolása viszont a jövőben még további kutatásokat és erőfeszítéseket kíván. Ennek egy jó iránya lehet a vázolt inkrementális SVM hatékonyabb megvalósításának alkalmazása, annotált videók automatizált felhasználásával.

Dolgozatom témájában cikk is készült, amely jelenleg elfogadásra vár.

9. Irodalomjegyzék

References

- [1] T. Ahonen, A. Hadid, and M. Pietikäinen. Face recognition with local binary patterns. *Computer Vision-ECCV 2004*, pages 469–481, 2004.
- [2] G.R. Bradski and A. Kaehler. *Learning opencv*. O'Reilly, 2008.
- [3] V. Bruce and S. Langton. The use of pigmentation and shading information in recognising the sex and identities of faces. *PERCEPTION-LONDON-*, 23:803–803, 1994.
- [4] R. Brunelli and T. Poggio. Hyperbf networks for gender classification. In *Proceedings of the DARPA Image Understanding Workshop*, pages 311–314, 1992.
- [5] G. Cauwenberghs and T. Poggio. Incremental and decremental support vector machine learning. In *Advances in neural information processing systems 13: proceedings of the 2000 conference*, page 409. The MIT Press, 2001.
- [6] B.A. Golomb, D.T. Lawrence, and T.J. Sejnowski. Sexnet: A neural network identifies sex from human faces. *Advances in neural information processing systems*, 3:572–577, 1991.
- [7] E. Hjelmås and B.K. Low. Face detection: A survey. *Computer vision and image understanding*, 83(3):236–274, 2001.
- [8] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *IEEE ICIP*, volume 1, pages 900–903. Citeseer, 2002.
- [9] E. Mäkinen and R. Raisamo. An experimental comparison of gender classification methods. *Pattern Recognition Letters*, 29(10):1544–1556, 2008.
- [10] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51–59, 1996.
- [11] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray scale and rotation invariant texture analysis with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.
- [12] L. Sajó, M. Hoffmann, and A. Fazekas. 3D head model from stereo images by self-organizing neural network. *Journal for Geometry and Graphics*, 13(2):209–220, 2009.

- [13] L. Sajó, Z. Ruttkay, and A. Fazekas. Turk-2, a multi-modal chess player. 2009.
- [14] R.E. Schapire and Y. Singer. BoosTexter: A boosting-based system for text categorization. *Machine learning*, 39(2):135–168, 2000.
- [15] G. Shakhnarovich, P.A. Viola, and B. Moghaddam. A unified learning framework for real time face detection and classification. In *International Conference on Automatic Face and Gesture Recognition*, pages 14–21. Citeseer, 2002.
- [16] C. Shan, S. Gong, and P.W. McOwan. Facial expression recognition based on Local Binary Patterns: A comprehensive study. *Image and Vision Computing*, 27(6):803–816, 2009.
- [17] N. Sun, W. Zheng, C. Sun, C. Zou, and L. Zhao. Gender classification based on boosting local binary pattern. *Advances in Neural Networks-ISNN 2006*, pages 194–201, 2006.
- [18] N.A. Syed, H. Liu, and K.K. Sung. Handling concept drifts in incremental learning with support vector machines. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 317–321. ACM, 1999.
- [19] JT Todd, LS Mark, RE Shaw, and JB Pittenger. The perception of human growth. *Scientific American*, 242(2):132, 1980.
- [20] P. Viola and M. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. In *Proc. IEEE CVPR 2001*. Citeseer, 2001.
- [21] H.A. Wild, S.E. Barrett, M.J. Spence, A.J. O’Toole, Y.D. Cheng, and J. Brooke. Recognition and sex categorization of adults’ and children’s faces: Examining performance in the absence of sex-stereotyped cues. *Journal of Experimental Child Psychology*, 77(4):269–291, 2000.
- [22] B. Wu, H. Ai, and C. Huang. Real-time gender classification. In *Proceedings of SPIE*, volume 5286, page 498, 2003.
- [23] M.K. Yamaguchi, T. Hirukawa, and S. Kanazawa. Judgment of gender through facial parts. *PERCEPTION-LONDON-*, 24:563–563, 1995.
- [24] M. Yang, B. Moghaddam, and B. Moghaddam. Gender Classification with Support Vector Machine. In *4th IEEE Intl. Conf. on Automatic Face and Gesture Recognition*, pages 306–311. Citeseer, 2000.

- [25] M.H. Yang, D.J. Kriegman, and N. Ahuja. Detecting faces in images: A survey. *IEEE Transactions on Pattern analysis and Machine intelligence*, pages 34–58, 2002.
- [26] P. Yang, S. Shan, W. Gao, S.Z. Li, and D. Zhang. Face recognition using ada-boosted gabor features. In *Proceedings of the 16th International Conference on Face and Gesture Recognition*, pages 356–361. Citeseer, 2004.

Internetes hivatkozások

OpenCV: <http://opencv.willowgarage.com/wiki/>

Libxml2: <http://xmlsoft.org/>

Pthreads: <http://sourceware.org/pthreads-win32/>

Visual C++ 2010 Express: <http://www.microsoft.com/express/Downloads/>

FERET database: http://www.itl.nist.gov/iad/humanid/feret/feret_master.html

Cohn-Kanade database: http://vasc.ri.cmu.edu/idb/html/face/facial_expression/

10. Köszönetnyilvánítás

Köszönet illeti témavezetőmet, dr. Fazekas Attilát segítségéért, konzulensemét, Sajó Leventét a segédanyagok összegyűjtésében nyújtott munkájáért és javaslataiért, továbbá munkatársamat, Szeghalmy Szilviát a dolgozatommal kapcsolatos építő és értékes észrevételeiért.